UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,327	10/31/2003	Cameron Beccario	MSFT-2768/305786.01	2566

41505 7590 03/18/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
03/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/699,327	BECCARIO ET AL.	
	Examiner	Art Unit	
	TED T. VO	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 December 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed on 12/31/2008.

Claims 1-27 are amending and pending in the application.

Response to Arguments

2. This action is in response to the amendment filed on 12/31/2008.

Applicants' argument remarks submit that their claimed limitation is not disclosed by C# language specification.

However, the limitation recited as

In a compiler, a method of determining a target type in an expression comprising an operator and at least one expression operand in a loosely-typed programming language, comprising steps of:

determining as said target type a most encompassed type from among a first set of types of loosely-typed operands, where said first set of types comprises all resulting types of all first variant expressions, where each of said first variant expressions comprises said target expression with at least one of said expression operands, at least one expression operand being a user-defined type, replaced using widening type conversion, if said first set is not empty;

if said first set is empty, determining as said target type a most encompassing type from among a second set of types of loosely-typed operands, where said second set of types comprises all resulting types of all second variant expressions, where each of said second variant expressions comprises said target expression with at least one of said expression operands, replaced using at least one of widening and narrowing type conversion; and

assigning said target type to said operator.

The method as a whole is for determining *a target type in an expression* which comprises an operator; upon the determination for relating *a most encompassed type,* and using various terminological languages that are only in a certain programming languages to understand, it assigns the target type to the operator. Therefore, it would not provide an ordinary to determine the claimed scope when considering "infringement".

On the other hand, the claim fails to be direct to a practical manner, but merely within a programming language. The claim fails to cause a real world transforming, but does the same like type resulting in the C# language specification. It should be noted that each programming language provides its own syntax. Claiming an infringement on its own programming language will prevent a user using the language specification to write programming; thus that programming language is useless. For example, if one claims C# or programming construct in C# to be an invention then no one will use C#.

Examiner has reviewed the Applicants' arguments, but they are generic. For example, Applicants depict the cited portions in p. 98-100, where applicants refer the portions teach user-defined conversion that is user, a programmer, defines a way to convert a variable from the first

type to a second type; Applicants generally allege that the cited portions are in contrast to their conversion that is for the compiler to do for a user or a programmer. The Applicants' arguments appear admitting the functionality for determining target type in the cited portions and the functionality of the Applicants compiler are the same, but their claim provides automatically; i.e. the assignment done by a compiler. It should be noted that the depicted citations show the determination of a target type in an expression, and show how to provide the type assignment for addressing the functionality of the claims. Therefore, even its assignment is by user's determination, it would not be different if the result would be generated by a compiler. The C# specification discusses for how the types be resulted in its syntax; therefore, when a C# compiler is carried out to the related programs, it will do assignment the same, i.e. the C# compiler designer has to convert the types as applicants argued.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. The claims 1-9 are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

As per Claims 1-9:

It should be noted that the claims had previously rejected under this statutory. After Applicants amending, the method is in a compiler, the Examiner withdrew the rejection.

The rejection to claims 1-9 will be rejected based on the recent decision in that a claim to qualify as § 101 statutory process must be tied to another statutory class, or positively recite the subject matter that is being transformed, for example by identifying the material that is being changed to a different state. (emphasis added) (See *In* re *Bilski,* 545 F.3d *943,* 88 USPQ2d 1385 (Fed. Cir. 2008). The court's opinion clarified the standards applicable in determining whether a claimed method constitutes a statutory "process" under § 101) .

The claims 1-9 recite merely instructions performed in programming. Even the method is done in a compiler, a compiler is a program. See each of the limitations in the claims; the process/method fails neither to be tied to another statutory class nor to transform underlying subject matter to a different state or thing. These methods/process claims fail to be statutory under 35 USC 101.

Suggestion: Amending the claimed scope as "A method used by a compiler stored in a computer system, when running by the computer processor, performing for determining…", for being statutory.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-27 are rejected under 35 U.S.C. 102(b) as being anticipated by "C# Language Specification", Version 0.28, May 2001 (hereinafter: C# Language Specification).

As per claim 1: The C# Language Specification discloses:

In a compiler, a method of determining a target type in an expression comprising an operator and at least one expression operand in a loosely-typed programming language, comprising steps of:

determining as said target type a most encompassed type from among a first set of types of loosely-typed operands (see discussing of determination of the most encompassed type from among a set of types in sec. 6.4.2, p.98-99), *where said first set of types comprises all resulting types of all first variant expressions, where each of said first variant expressions comprises said target expression with at least one of said expression operands, at least one expression operand being a user-defined type, replaced using widening type conversion* (See incorporation of the reference in the type conversion: Sec. 6.4, start at p. 98), *if said first set is not empty;*

if said first set is empty, determining as said target type a most encompassing type from among a second set of types of loosely-typed operands (see discussing of determination of the most encompassed type from among a set of types in sec. 6.4.2, p.98-99), *where said second set of types comprises all resulting types of all second variant expressions, where each of said second variant expressions comprises said target expression with at least one of said expression operands, replaced using at least one of widening and narrowing type conversion* (See incorporation of the reference in the type conversion: Sec. 6.4, start at p. 98)

(Note: The above claim is anticipated by rules set for in the C# programming specification. See p. 98-100, For "*determining*", see the definitions for most encompassed types among set of type A, and most encompassing type among set of type B (p. 99, the seventh ??, eighth ?? and ninth ??); and the discussions of user-defined implicit conversions and user-defined explicit conversions. These definitions/discussions have the means for determining a type as a most encompassed type from among a set of types and the means for determining a type as a most encompassing type from among a set of types. These are incorporated in the teachings disclosed in p. 107. See p. 107: in sec. 7.2.4, where the set of candidate user-defined operator (*first set*) provided by X and Y ("*expression operand being a user-defined type, replaced using widening type conversion*" reads on union, provided by of X and provided by Y), and see in 7.2.4, first ?? : using rules 7.2.5, or see second ??: If this set is not empty, then…, : "*determining as said target type a most encompassed type from among a first set of types of loosely-typed operands*". Also see in the second ?? and third ??: 'Otherwise, the predefined operator op implementations (*second set*) become the set of candidate operators for the operation ', and the acts thereafter: "*determining as said target type a most encompassing type*

from among a second set of types..". Variables such as x, or y in the expressions: *variant expressions*); *and*

assigning said target type to said operator.

As per claim 2: Claims 2 recite the functionality of claim 1 in determining the first set of types, but further defining operands that comprising n operands where each operand O_m is a specific type T_m.

See the rationale in Claims 1, and see the section 7.2.4, it discusses at least operand x in the set X or y in the set Y for user-defined operators. Assume that X is not empty and has more than one element. For example, n elements

As per claim 3: Claims 3 recite the functionality of claim 1 in determining the second set of types, but further defining hypothetical operands that comprising n operands (n+1 to n+n) where each operand HO_{n+m} is a specific type T_m. With regards to the limitation recited in Claim 3, see the rationale in Claims 1, and see the section 7.2.4, it discusses at least operands list (x,y) that became result of overload resolution process.

As per claim 4: Claim 4 recites the converting the operands and computing which read on the conversion process of types in the reference.

As per claims 5-7: The limitations recited in the claims are intended uses, i.e. the limitations are included without being functioned in the claims. Thus, see operands/operators/type in the reference for disclosing these claims.

As per Claims 10-16: See rationale addressed above for Claims 1-7.

As per Claims 19-25: See rationale addressed above for Claims 1-7.

As per Claim 8: C# Language Specification discloses:

In a compiler, a method of resolving an expression comprising an overloaded binary operator (See table in p. 105: included with binary operators), ***a first operand of a first type and a second operand of a second type,*** *in a loosely-typed programming language,* (See depicted operation x op y: x may be an operand of first type, and y maybe operand of a second type (an example in the last 3-lines in p. 107 to an operation such as b * s, where b is a byte and s is short, etc)), ***comprising steps of:***

parsing the expression to determine said overloaded binary operator, the first operand and the second operand, the first operand and the second operand each being loosely typed, at least one of the set comprising the first operand and the second operand being a user-defined type (See all sections of 7.2.6, start at p. 107)***;***

determining a first set of types, where said first set comprises all types to which there is a widening conversion from said first type; (depict an example of b * s, the first set may be includes widening conversion members of all byte types, such as "int" – See sec. 6, for "conversion" and see using the set of type A or type B)

determining a second set of types, where said second set comprises all types to which there is a widening conversion from said second type (depict an example of b * s, the second set may be includes widening conversion members of all short type, such as "int" - See sec. 6, for "conversion" and see using the set of type A or type B)

There are many sets of types in the reference, for example, see sec. 6.4.2. For example set of type A or type B, type S, type X, and T, etc.

determining a third set of types, where said third set comprises all types which result from an operation of said overloaded binary operator on a type from among said first set and a type from among said second set;

See each determination of Conversions in sec 6, where the third set of types meets all conditions of a depicting type: For example(p.96: lines:1-4): If a set of type is char then result sets are sbyte, byte, or short – i.e. all the result types in a standard conversion to a target type (in p. 99)). And then see sec. 7.2.4, start at p. 107, a generic candidate binary operation op (x, y) of a user defined operator, the third set is the set of result types in overload resolution process, where it is incorporated from the result types for the types in X and the types in Y as discussed in sec.6.

if said third set of types is empty, determining a fourth set of types, where said fourth set comprises all types to which there is a narrowing conversion from said first type and all types to which there is a widening conversion from said first type;

i.e. the result types of x in X and of y in Y do not contain a type for both (similarly to the example of b * s, the result type is not empty), then the fourth set reads on the result types for x in X, with respect to user candidate binary operator of sec. 7.2.4, as it is incorporated to sec .6 for type conversion.

if said third set of types is empty, determining a fifth set of types, where said fifth set comprises all types to which there is a narrowing conversion from said second type and all types to which there is a widening conversion from said second type;

i.e. the result types of x in X and of y in Y do not contain a type for both; then the fifth set reads on the result types for y in Y, with respect to user candidate binary operator of sec. 7.2.4, as it is incorporated to sec .6 for type conversion.

if said third set of types is empty, determining a sixth set of types, where said sixth set comprises all types which result from the operation of said overloaded binary operator on a type from among said fourth set and a type from among said fifth set;

i.e. the union of result types of x,y under the type overload in sec. 7.2.4, as it is incorporated to sec .6 for type conversion.

if said third set of types is not empty, selecting the most encompassed type in said third set as a target type; See sec 6, for determining the most encompassed type;

if said third set of types is empty, selecting the most encompassing type in said sixth set of types as said target type See sec 6, for determining the most encompassing type; ***and***

assigning said target type to said overloaded binary operator (i.e. type conversion for a binary operator in the overload resolution).

As per Claim 9: C# Language Specification discloses,

converting said first operand to said target type;

converting said second operand to said target type;

computing said operation <u>of said overloaded binary operator</u> on said converted first operand and said converted second operand. See overload resolution process as discussed, start from sec. 7.2.4, and its incorporation of sec. 6.

<u>As per Claims 17-18</u>: Refer to the rationale as addressed top Claims 10-16.

<u>Claims 26-27</u>: Claims 26-27 is merely manipulating a mathematic algorithm.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
March 13, 2009

/Ted T. Vo/
Primary Examiner, Art Unit 2191